Document Flow Segmentation for Business Applications

Hani Daher[a] and Abdel Belaïd[b*]

[a] LORIA, Campus scientifique, Vandoeuvre-Lès-Nancy, France;
[b] Université de Lorraine-LORIA, Campus scientifique, Vandoeuvre-Lès-Nancy, France

ABSTRACT

The aim of this paper is to propose a document flow supervised segmentation approach applied to real world heterogeneous documents. Our algorithm treats the flow of documents as couples of consecutive pages and studies the relationship that exists between them. At first, sets of features are extracted from the pages where we propose an approach to model the couple of pages into a single feature vector representation. This representation will be provided to a binary classifier which classifies the relationship as either segmentation or continuity. In case of segmentation, we consider that we have a complete document and the analysis of the flow continues by starting a new document. In case of continuity, the couple of pages are assimilated to the same document and the analysis continues on the flow. If there is an uncertainty on whether the relationship between the couple of pages should be classified as a continuity or segmentation, a rejection is decided and the pages analyzed until this point are considered as a "fragment". The first classification already provides good results approaching *90%* on certain documents, which is high at this level of the system.

Keywords: Document Flow segmentation, Textual descriptors, Business flow, Continuity and rupture classification

1. INTRODUCTION

Various types of documents flow into organizations every day, form claims, forms, invoices, contracts and more. Handling this flow of information manually by sorting the documents is a time consuming, costly and error-prone task. One solution is to introduce page separators or machine readable marks like bar codes to indicate the end of a document.

In the case of page separators this approach is costly and intensive because they must be inserted before the scanning of pages and, if they are not to be reused, removed afterwards. In high volume operations, these costs can be staggering. In the case of bar codes they offer more accurate document identification, but also at high cost. The use of papers, ink and codes is not an easy task and it also costly. Inserting these bar codes between documents is error prone and must be inserted correctly to ensure that the correct separator sheet is used.

The objective of our work is to develop an automatic segmentation approach capable of segmenting a stream of documents, without the need of any prior knowledge on the number of pages or on the document class, and where each document may represent a set of successive well-ordered pages.

Furthermore, we should take into consideration that we are dealing with a heterogeneous flow of multipage documents where the quality of pages that constitute the documents may vary, and where some information may be accessible in one document but not the other.

This paper is organized as follows. In section *1.1* we present the state of the art by highlighting it into three categories. In section *2* we describe our approach, finally in section *3* we show the results and experiments.

1.1 Literature Review

To our knowledge, very few methods have been proposed to tackle this subject and find solutions. In our research we identified three categories of approaches used in document flow processing:

- Document Segmentation: where the task is to partition a flow of documents into multiple subsets of documents.
- Document Retrieval: where the task is to search a database for the closest images to a query image
- Document Classification: where the task is to assign a document to one or more classes or categories

We are going to illustrate in the following sections these different cases and point out the closest elements to our topic. Table *1* summarizes all the works discussed below.

1.1.1 Document segmentation

Collins-Thompson and Nickolov [1] work on page similarity which relies on structural and textual similarities. The authors treat document separation as a bottom-up clustering problem, where every page is considered as a cluster, and then proceed in steps by merging pairs of clusters using a single-linkage criterion. In the proposed method, page numbers are considered as always located in the bottom of the page which is not always the case. If these features are not correctly localized; this will affect the classification result. Textual and visual features are also extracted from the documents. Results show that the combination of the visual and textual features produces a segmentation accuracy of *95.68%* and the visual features alone an accuracy of *89.25%*. The authors consider that pages in a same document contain a lot of similarities, which is not always the case; in real world applications the content of pages may bear very little similarities. The method proposed by Meilander and Belaid [2] is similar to the variable horizon models (*VHM*) or multi-grams used in speech recognition. It consists in maximizing the flow likelihood knowing all the Markov Models of the constituent elements. As the calculation of this likelihood on all the flow is *NP*-complete, the solution consists in studying them in windows of reduced observations. The first results obtained on homogeneous flows of invoices reaches more than *75%* of precision and *90%* of recall. The method was only tested on homogeneous documents and the proposed model is only suited for the invoices class.

1.1.2 Document Retrieval

Rusiñol et al. [3], study different approaches for multipage document retrieval. They propose two fusion strategies including the early and late fusion to form one document representation out of a set of pages. Two types of features extracted from the pages of documents are visual and textual. The visual features are based on the *SIFT* descriptors and textual features are represented by bag of words that are weighted by the *tf-idf* method. The results show that the textual features gave good results by using the late fusion technique with an accuracy of *74.24%*. Visual features on the other hand didn't give good results with an accuracy of *47.74%*, this low accuracy is due to the fact that the documents were more oriented semantic, and the structure of the document didn't offer much information. Many documents in the same class share the same subject but are physically different. Kumar et al. [4] presented a bag of words approach using a feature pooling strategy. The algorithm was tested respectively on two types of documents, forms and tables, with an accuracy of *97.4%* and *98.9%*. This method relies heavily on the structure of the document, and is applied on documents of single pages, which is not our case, where we have to deal with multipage documents, and where the structure yields so little information about the classes of documents. Shin et al. [5] segment the pages into blocks that are characterized by conceptual and geometric features. The distance between the query image and the other images in the database is achieved by mapping the blocks of the images, with a retrieval accuracy of *89%*. The drawback of this method is that the extracted features are very specific to a category of documents.

1.1.3 Document classification

Gordo et al. [6] treated the problem of multipage documents classification. Every document is composed of different classes of pages including: papers, insurance, invoice, etc. By using the same principle of the bag of words, they propose a bag of pages approach where a document is represented by a histogram that includes the number of occurrences of these classes. The method was tested on two datasets of different sizes. The results show that by increasing the number of documents, the classification results decreased drastically with a difference of *21.1%* between the first and the second database. Shin et al. [7] proposed a document classification algorithm based on visual similarity of layout structure.

Features such as column structures, percentage of text and non-text are extracted. This method does not propose a strategy for multipage documents classification. It is well adapted for single page documents.

The previous works show that the textual features did outperform the visual features in terms of classification and retrieval accuracies. The combination of the two types of features did improve the results but it wasn't enough, computational time was added to the algorithms. Furthermore, the database that we are working on relies more on the content of documents rather on the structure, so adding an extra complexity to our approach won't help us solve the problem. The fusion techniques of features are very interesting and we inspired our method from these approaches as described in section *2*.

Table1. Summary of the discussed methods. S = Segmentation, C = classification, R = Retrieval, T = Textual features, V = Visual features

Authors(s)	***Method***			***Features***		***Base***		***Precision (%)***		
	S	***C***	***R***	***T***	***V***	***Documents***	***Pages***	***T***	***V***	***T+V***
Collins-Thompson and Nickolov [1]	•			•	•	*191*	*2709*		*89.25*	*95.68*
Meilander and Belaid [2]	•			•		*356*	*719*	*76*		
Rusiñol et al. [3]			•	•	•	*7 200*	*70 000*	*74.24*	*47.74*	
Kumar et al. Ref. [4]			•				*1411*		*98.9*	
Shin et al. Ref. [5]		•			•		*979*		*89*	
Gordo et al. Ref. [6]		•			•	*21 238*	*67 627*		*54.4-75.5*	
Shin et al. Ref.[7]		•			•		*5590*		*99.70*	

2. PROPOSED APPROACH

Figure *1* illustrates the three main modules of the proposed approach which are: feature extraction, relationship modeling of a pair of pages and classification. These three modules will be explained in details in the next sections.

All images are OCR-ed and stop words have been removed, as output every page will be presented by an *XML* tree which is composed of a set of blocks $\{B_i\}$. Every block includes a sequence of sections $[S_n]$. Every section includes a sequence of words $[w_{k\neq l}]$ which are the root nodes. Every block, section and word, are given by the coordinates of their bounding boxes *named (top, left), (bottom, right).*

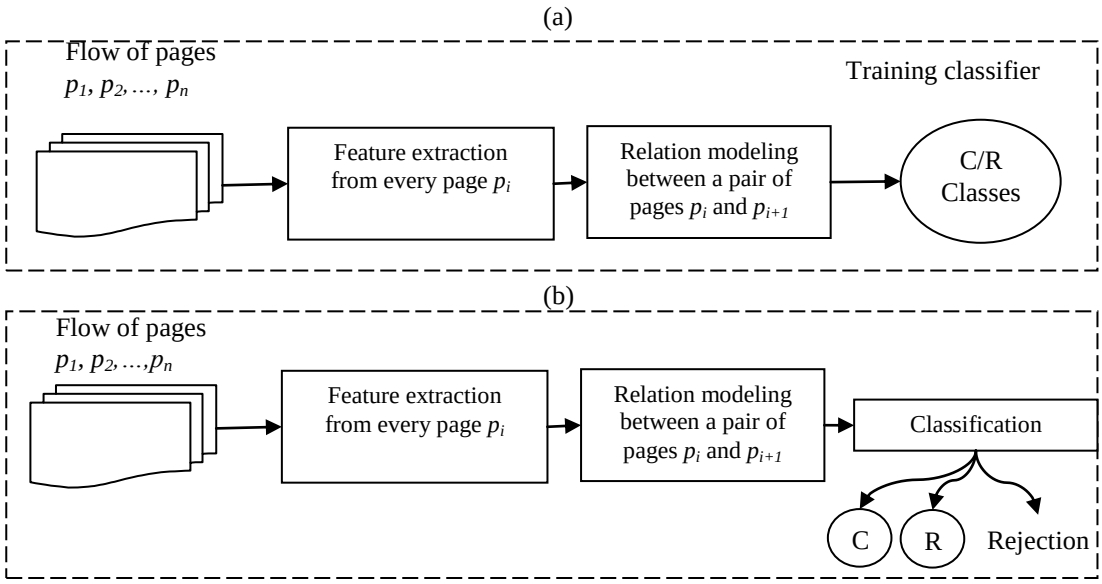


Figure 1. Document segmentation flow chart, based on supervised classification. (a) training, (b) testing

2.1 Feature extraction and relationship modeling

We start by analyzing what are the elements that are found in the business documents and that might help us identify if there is a potential continuity or rupture between two consecutive pages. We only focused on textual elements and we classified them into *6* classes according to Table 2.

Table 2. Classes of features that were extracted from the business documents

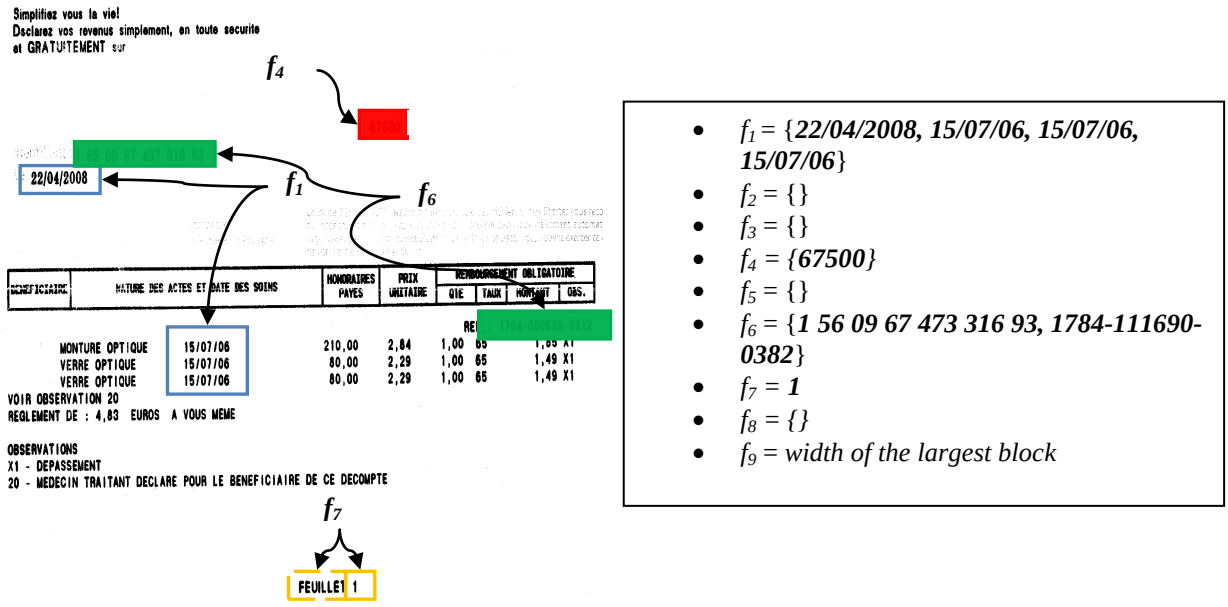
Fax	**Page**	**Date**	**Code**	**Number**	**ID**
Date	Number	Expedition	Account	Number	Global ID
Number	Font	Assignation	Receiver	Folder	User ID
Page Number	Margin	Mission	Shipper	Social Security	
Fax	Item	Deadline	Immatriculation	Client	
Hour	Logo	Invoice	Zip Code	Order	
	Signature	Report	Reference	Contrat	
	Sequence	Transaction	Commercial Reference	Transaction	
	Salutation	Exchange	Transaction Reference	Invoice	
		sinister	TVA	Tax	
		Accident	Pin	General	
		Internal		Licence	

To be generic and in order to cover all the descriptors, we reduced these features to a set of *9* main features. All dates are represented by a single feature f_1, modeling the different date formats. The hour f_2 represents all the hour formats; the telephone f_3 is suited for all the telephone formats in France. The Zip code f_4 extracts all the French Zip codes of length *4*. The alphanumeric feature f_5 extracts all the alphanumerical patterns. The numeric feature f_6 extracts all the numbers with a length greater than *5*, this way the extraction algorithm won't confuse the numeric features with the Zip code. Page numbers f_7 represents the number of the page, in contrast to the other features that are extracted by value, the page number is identified by its label (like page…), then the page number located on the right is extracted. The Salutation feature f_8 combines all the French salutations. The Margin f_9 represents the width of the largest *block* and is computed as follow: $f_9 = |right - left|$.

Table 3. Features representation

Feature	Description	Type
f_1	Date ***(d)***	Alphanumeric String
f_2	Hour ***(h)***	Alphanumeric String
f_3	Telephone ***(t)***	Numeric string
f_4	Zip Code ***(z)***	Numeric string ***length = 4***
f_5	Alphanumeric ***(a)***	Alphanumeric String
f_6	Numeric ***(n)***	Integer string ***length > 5***
f_7	Page Number ***(p)***	Numeric
f_8	Salutation ***(s)***	String
f_9	Margin ***(m)***	Numeric

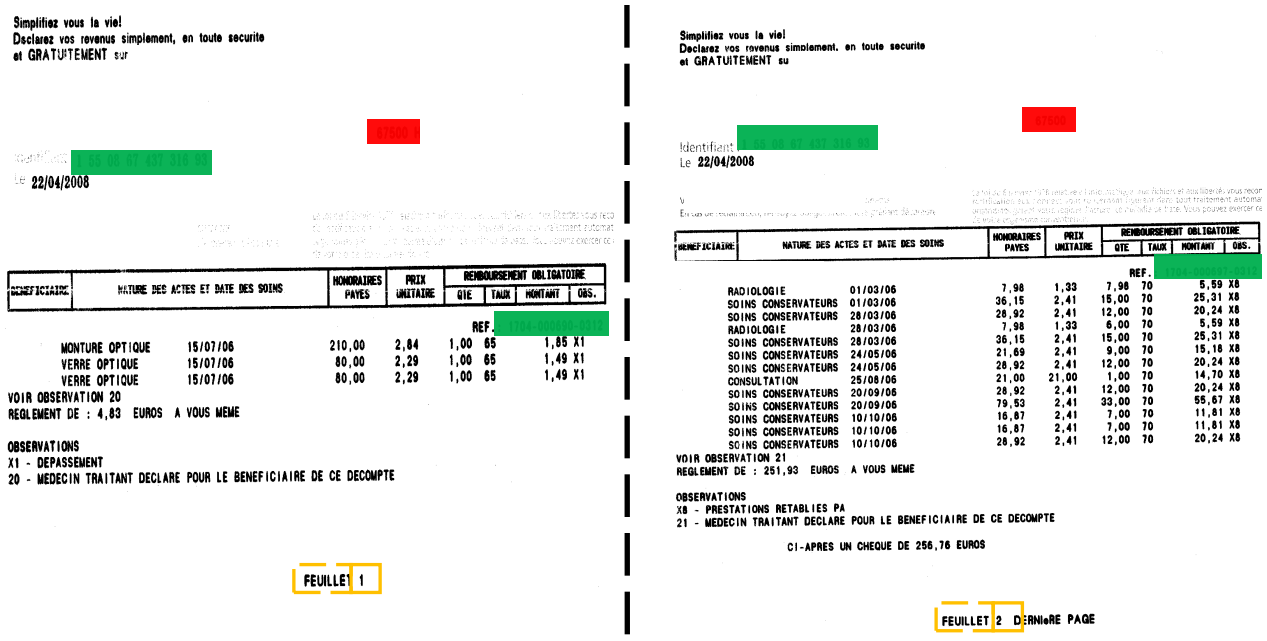
Features $f_1,\ldots,f_8$ are extracted by regular expressions. All the features except *f7* and *f9* represent the set of all the values related to their type and are found by regular expressions.



- f_1 = {***22/04/2008, 15/07/06, 15/07/06, 15/07/06***}
- f_2 = {}
- f_3 = {}
- f_4 = {***67500***}
- f_5 = {}
- f_6 = {***1 56 09 67 473 316 93, 1784-111690-0382***}
- f_7 = ***1***
- f_8 = {}
- f_9 = *width of the largest block*

Figure 2. Example of feature extraction by regular expressions

f_1 is composed of a set of *3* dates, $f_1 = \{d_1,...,d_3\}$. The regular expression didn't find any pattern related to feature f_2. We consider then that f_2 represents an empty set, $f_2 = \{\}$. f_7 is represented by *1*, the regular expression found the label *"Feuillet"* and assigned the value *1* located on its right side to f_7 (See Figure 2). In order to model the relationship between two consecutive pages p_i and p_{i+1}, and to identify the continuity and rupture, we have to find a relation *R* between the couple between of pages p_i and p_{i+1} $C(p_i, p_{i+1})$.



Page $_i$

- f_1 = {***22/04/2008, 15/07/06, 15/07/06, 15/07/06***}
- f_2 = {}
- f_3 = {}
- f_4 = {***67500***}
- f_5 = {}
- f_6 = {***1 56 09 67 473 316 93, 1784-111690-0382***}
- f_7 = ***1***
- f_8 = {}
- f_9 = *width of the largest block*

Page $_{i+1}$

- f_1 = {***22/04/2008, 01/03/06,…,10/10/06***}
- f_2 = {}
- f_3 = {}
- f_4 = {***67500***}
- f_5 = {}
- f_6 = {***1 56 08 67 437 316 93, 1708-000690-0316***}
- f_7 = ***2***
- f_8 = {}
- f_9 = *width of the largest block*

$$v_i = \{f_1 = \{d_1,...,d_4\},..., f_6 = \{n_1,...,n_2\}, f_9\} \qquad v_{i+1} = \{f_1 = \{d_1,...,d_{14}\},..., f_6 = \{n_1,...,n_2\}, f_9\}$$

$$R(v_{ci}) = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ \hline f_1 & f_2 & f_3 & f_4 & f_5 & f_6 & f_7 & f_8 \\ \hline \end{array}$$

Figure 3. Relationship modeling procedure

Figure 3, shows how $R(v_{ci})$ is constructed. In the case of f_1, since at least one value from *page i* is equal to the value of *page i+1* the first feature in $R(v_{ci})$ is attributed the value *1*. f_2, f_3, f_5 and f_8 are not present in both of the pages so they are attributed the value *0*. For the feature f_7 that represents the page number, since the value in *page i* is smaller than the value in *page i+1* then it is attributed the value *1*.

- *-1:* implies that the features exist but are different, reflecting a segmentation or rupture
- *0:* There is no correspondence between the features or the features does not exist, reflecting a potential rejection
- *1:* implies that there are plenty of ties between the features, reflecting a continuity
- For feature f_7 the comparison is done by value independently from the label which is only used to identify the page number. If $f_{7vi} \prec f_{7vi+1}$ then we have a value *1* else *-1*
- For feature f_9 the Euclidean distance is used to compute the difference between the widths of the margins. If the distance is less than a threshold $\delta = 12$ we consider that there is a continuity between the couple of pages *d(.,.) = 1*, else *d(.,.) = -1*

2.1.1 Formalisation

Let $S = \{p_1,..., p_n\}$ be the stream of pages. Every page p_i is represented by a vector v_i of dimension *9*.

$$v_i = \{f_1 = \{d_1,..., d_k\} \vee \{\}, ..., f_8 = \{s_1,..., s_n\} \vee \{\}, f_9\} / k \neq n$$

v_i is a vector of vectors, every vector can be either empty or may represent a set of patterns *p*. Every vector may differ from the other by its dimension. For every feature f_{vi} except features f_7 and f_9 represented by a set of extracted patterns, we compute the intersection with the same feature f_{vi+1} of the successive page. The intersection presents the equality between the patterns of a set. The value assigned to the intersection is an integer which takes multiple values, *"1"* if there is at least one pattern in f_{vi} that intersect with another patter of f_{vi+1}, *"-1"* if there is no intersection, and *"0"* is both the sets of patterns or one of them in f_{vi} and f_{vi+1} is empty $\{\}$.

$$R(v_{ci}) = v_i \cap v_{i+1} = \{f_{1vi} \cap f_{1vi+1}, ..., f_{7vi} \prec f_{7vi+1}, f_{8vi} \cap f_{8vi+1}, d(f_{9vi}, f_{9vi+1})\}$$

$$\begin{cases} (\{\} \cap \{\}) \Rightarrow v_{ci} = 0 \\ (\{\} \cap \{p'_1,..., p'_j\} \vee \{p_1,..., p_j\} \cap \{\}) \Rightarrow v_{ci} = 0 \\ (\{p_1,..., p_j\} \cap \{p'_1,..., p'_l\}) \neq \phi \Rightarrow v_{ci} = 1 \\ (\{p_1,..., p_j\} \cap \{p_1,..., p'_l\}) = \phi \Rightarrow v_{ci} = -1 \end{cases}$$

2.2 Classification

As input the classifier will take the vectors $R(v_{ci})$ representing the couple of pages. For the sake of clarification we replace $R(v_{ci})$ by *x*. the classifier classifies each incoming couple of pages as belonging to the same document; (continuity *(1))* or not (rupture *(0)).* As we stated earlier, we added an extra layer to the decision of the classifier based on the class membership probabilities which reflects the uncertainty with which a given couple of pages can be assigned to any given class, and is represented by an evaluation function *E* working as follows:

$$E(x) = |P(\hat{c}|x) - P(c|x)|$$

$$\begin{cases} E(x) \succ \sigma & 0 \vee 1 \\ else & Over-segmentation \end{cases}$$

P(c | x) is the probability of *x* belonging to class *c*. If $E(x) < \sigma = 0.6$, then we have a case of uncertainty leading to a fragment of a document. The over-segmentation choice is based on the assumption that we might end up with missing pages of a document but we will never end up with pages that belong to different classes of documents being fused into one.

3. EXPERIMENTS

All of our experiments were carried on databases provided by ITESOFT Company (see Figure 3). To test the stability of the approach, we used four databases containing different numbers of documents and pages. The first database contains very heterogeneous documents that can be easily separated by the classifier. The other three databases are also heterogeneous but with classes of documents that cannot be separated so easily. Some classes of documents might contain features similar to other classes in the same database. On all the databases *75%* of the documents were used for training and *25%* for testing.

- Database *1* : *618* documents (*2366* pages)
- Database *2* : *1898* documents (*7405* pages)
- Database *3* : *802* documents (*11759* pages)
- Database 4: *3318* documents (*21530* pages)

Table 4. shows the results of our segmentation approach on the *4* databases.

Table 4. Segmentation results

Classifier	Precision	Recall	F-measure
Database 1			
Voted Perceptron	*0.95*	*0.95*	*0.95*
SVM	*0.94*	*0.94*	*0.94*
Multilayer Perceptron	*0.94*	*0.94*	*0.94*
Multi-Boost	*0.92*	*0.91*	*0.91*
Database 2			
Voted Perceptron	*0.80*	*0.81*	*0.79*
SVM	*0.80*	*0.80*	*0.75*
Multilayer Perceptron	*0.80*	*0.79*	*0.79*
Multi-Boost	*0.81*	*0.81*	*0.81*
Database 3			
Voted Perceptron	*0.79*	*0.80*	*0.77*
SVM	*0.76*	*0.77*	*0.71*
Multilayer Perceptron	*0.80*	*0.81*	*0.80*
Multi-Boost	*0.79*	*0.81*	*0.79*
Database 4			

Voted Perceptron	*0.80*	*0.81*	*0.80*
SVM	*0.81*	*0.81*	*0.80*
Multilayer Perceptron	*0.81*	*0.82*	*0.81*
Multi-Boost	*0.81*	*0.81*	*0.79*

The classifiers produce good results on the first database, since the document classes are quite different and can be easily segmented. The precision and Recall values remained stable even when added an extra complexity presented by the increase of the number of documents and pages.

Figure 4 shows the stability of our system even after increasing the size of documents and pages in the database.

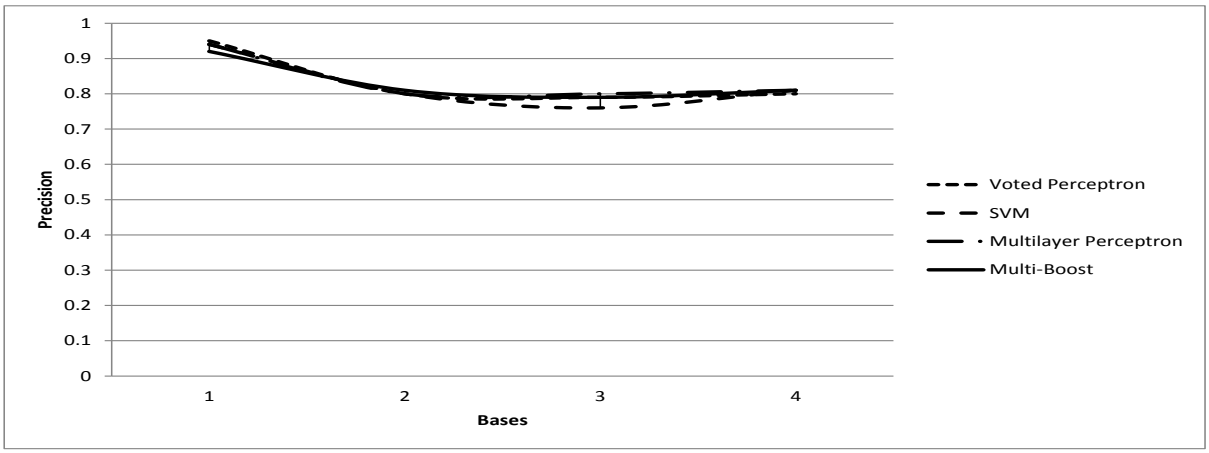


Figure 4. Precision curve

3.1 Discussion

In order to understand what caused the segmentation errors, we analyzed the cases of rupture and continuity separately on database *4* since it is the largest database. For the case where ruptures were classified as continuities we count the occurrence of descriptors indicating continuity. The same analysis was carried out on the continuity errors.

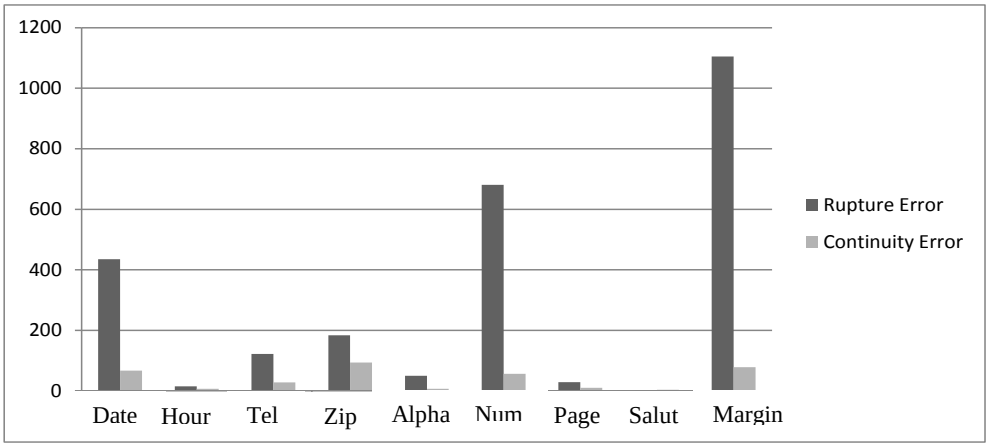


Figure 5. Continuity and rupture errors

Figure 5 shows that feature f_1, f_6 and f_9 are the ones that appear the most in the case where we have rupture errors. The same goes for the continuity errors. We also notice that extracted features favor continuity over segmentation since the rupture errors are more present than those of the continuity.

We also notice that the other features have no weight and are not discriminant. In fact discarding those features have a very little effect on the classification results (see Figure 6). When we remove the feature with the higher weight, we see that the classification accuracy drops.

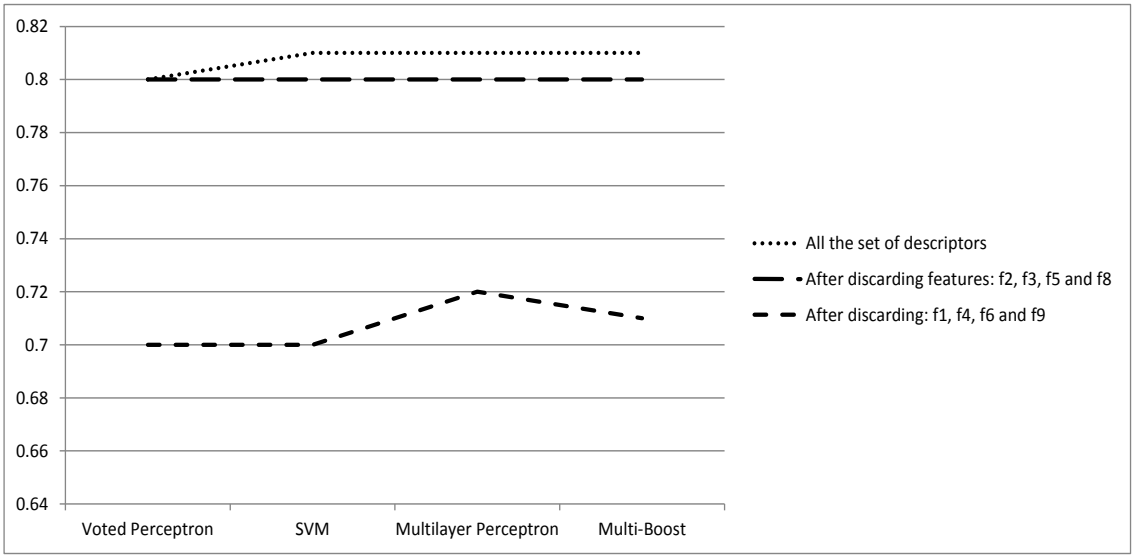


Figure 6. Precision after discarding specific subsets of features

Figure 7 illustrates the proportion of fragments on the *4* databases on the different classifiers. The SVM and Voted perceptron produce *0* fragments; there was no uncertainty in the classification. In the 2^{nd} database documents are very close in semantics this why the multilayer perceptron produces *29%* of fragments.

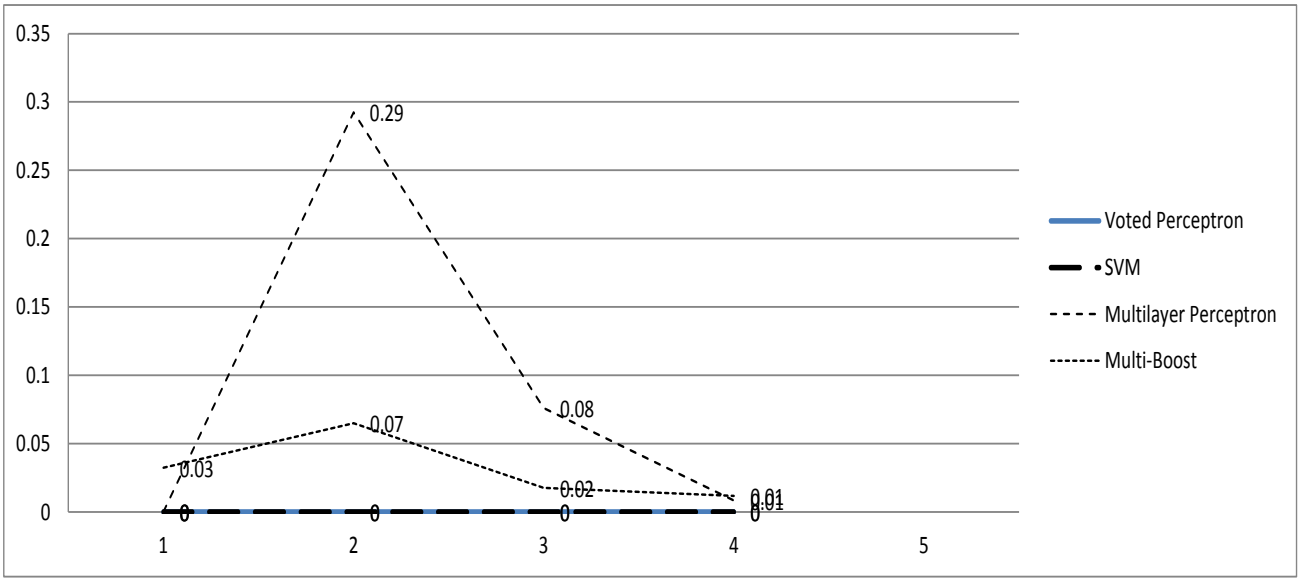


Figure 7. Proportion of fragments per Base

4. CONCLUSION

We proposed in this article a generic approach for the segmentation of a heterogeneous flow of documents. Results show the stability of our approach. The increase of the number of documents didn't affect the results. This study allowed us also to measure the effectiveness of the features and their discriminating power. The second step is the verification. The fragments obtained by this our approach represent ambiguous cases where there is probability of a segmentation error. The verification algorithm will read the sequence of fragments and assign to each fragment a class of a document (invoice, insurance etc.), if the confidence probability is less than a threshold then the classification is correct, else the fragment will be sent to the Case Based Reasoning module where its role will be to find a solution to the problem presented by the sequence of fragments. As output, this module provides a solution represented by a sequence of documents. The final objective of our study is the fusion of documents that are similar.

REFERENCES

1. Collins-Thompson, K. and Nickolov, R., "A clustering-based algorithm for automatic document separation," SIGIR, (2002).
2. Meilender, T. and Belaïd, A., "Segmentation of continuous document flow by a modified backward-forward algorithm," Proc. SPIE, (2009).
3. Rusiñol, M., Karatzas D., Bagdanov and Llados J., "Multipage document retrieval by textual and visual representations," ICPR, (2012).
4. Kumar, J., Ye, P. and Doermann, D., "Learning Document Structure for Retrieval and Classification," ICPR, (2012).
5. Shin, C. and Doermann, D., "Document image retrieval based on layout structural similarity," ICIP, (2006).
6. Gordo, A. and Perronnin, F., "A bag-of-pages approach to unordered multi-page document classification," ICPR, (2010).
7. Shin, C., Doermann, D. and Rosenfeld, A., "Classification of document pages using structure-based feature," IJDAR, (2001).